

BYTE

Reprinted with permission from the December 1990 edition of BYTE magazine.
Copyright ©1990 by McGraw-Hill, Inc., New York, 10020. All rights reserved.



INSPIRATION AT THE YEAR'S END

The Mac markets changed during 1990, and new markets are developing

How time flies. The first year of the new decade whipped by as fast as a screen update on a Mac IIcx. It's time to evaluate what's happened to the computing community, including the Mac's position within it. In the IBM-compatible world, the Rest of Them have been cranking out 486 systems at higher clock speeds, and Windows 3.0 is a major improvement as far as DOS-based graphical user interfaces (GUIs) go.

Apple, after some dithering about, took muster and waged the battle of the market share on two fronts. It took the high ground in March with its introduction of the Mac IIcx, the first-ever Mac clocked at 40 MHz. In October, Apple began the blitzkrieg on the low end with three new low-cost Macs. Just as important, the company slashed prices on the existing models.

Apple's attack plan so far has been impressive, but it's not without flaws. The Mac IIcx's much-touted SCSI DMA will not work with anything but A/UX. System 7.0 still hasn't seen the light of day. The features on the low-cost Macs are so-so, including their pricing. If Apple is indeed in the market-share battle for the long haul, I wonder whether it will have the nerve to reduce prices further if the situation warrants it. In other words, we've gone through another year of some improvements to the Mac and its software, but we're still waiting for the next Mac revolution. I'm getting pretty itchy waiting for it to come.

The one place I see a difference is not in the volleys from the big guns over-



head, but when I look in the trenches where the small Mac developers hunker down. Try as we might, we just can't kill these little guys. We entice them with promises of greater profits (Windows 3.0), hotter development systems (Next-Step), and other baubles, but they stick to the Mac like glue.

Whatever the reason, it's these folks who make the most interesting Mac software today, and who typically define new software categories. These developers think about more than just adding more features to a word processor or better kerning to a page-layout system. They think about what might be possible with a Mac.

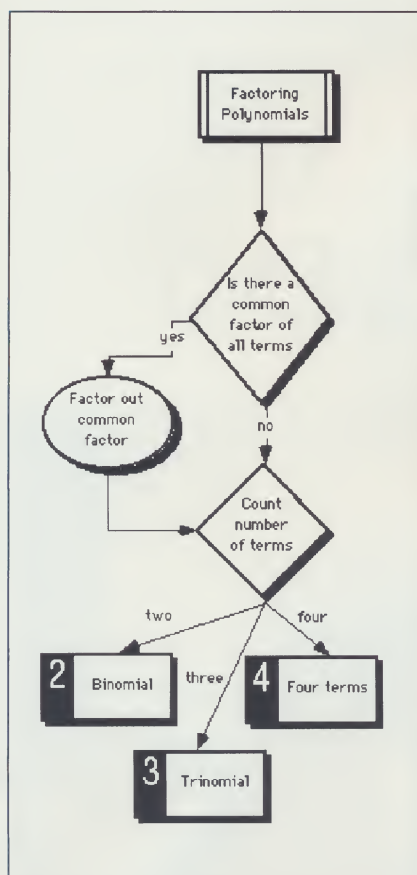
The Mac Gets Inspired

Not surprisingly, one of the newer Mac products I've found is called Inspiration 2.0. It's from a tiny family-owned company called Ceres Software in Portland, Oregon, and it's been around for a year

now, largely ignored. I spent a pleasant time conversing with its creator, Don Helfgott. He calls his program a thought processor, but that doesn't do it justice.

Inspiration 2.0 takes the Mac's ability to give you a consistent and meaningful GUI and lets you brainstorm, cranking out ideas by the bushel basket. It then organizes the ideas in a way that helps you realize what, if anything, you've really invented. Inspiration 2.0 is sort of a visual diagramming and outlining tool interwoven with a simple word processor. It's the first computer aid to creative thinking I've tried that actually works—and I've tried them all.

The theory behind Inspiration 2.0 goes something like this. Databases, word processors, and spreadsheets store and access information in ways that don't help us relate information more clearly, because they fail to distinguish between right-brain thinking (i.e., creativity, images, comprehension, imagination,



Inspiration 2.0 combines visual diagramming and outlining with a simple word processor. This Inspiration flowchart outlines the general algorithm for factoring a polynomial, as shown in listing 1.

Listing 1: This listing was derived from the Inspiration flowchart shown in the figure.

```

Factoring polynomials
  Is there a common factor of all terms?
    Count number of terms
      Binomial
        Difference of two squares
           $a^2 - b^2 = (a + b)(a - b)$ 
        Sum of two cubes
           $a^3 + b^3 = (a + b)(a^2 - ab + b^2)$ 
        Difference of two cubes
           $a^3 - b^3 = (a - b)(a^2 + ab + b^2)$ 
      Trinomial
        Trial and error or AC method
      Four terms
        Factor by grouping
      Factor out common factor
  
```

visualization, music, and daydreaming) and left-brain thinking (i.e., logic, sequence, reasoning, and analysis). Inspiration 2.0 works by filling the gap between these two kinds of thinking.

Other programs have promised to accentuate the synergy between the right and left brains. Neil Larson's Maxthink, early versions of Dave Winer's Think-Tank outliner, ODS's Consultant, and a myriad of other shareware have all skipped through this software minefield before. None of them really pulled it off, however, because they just didn't have a good model to implement.

Inspiration 2.0 has a good model, and that's why it succeeds. Rather than trying to be all things to all thinkers, the Helfgotts have pinned down their customers—writers, entrepreneurs, creative directors, analysts, planners, educators, and consultants.

They did not make Inspiration 2.0 some kind of unwieldy general management tool, or imbue it with powerful features that its target audience will not need. In short, they didn't kitchen-sink this product by trying to make it fit the needs of a general presentation tool, a business graphics package, or a free-form charting tool.

The only thing that Inspiration 2.0 assumes is that you think visually. If you never leave the mental domain of numbers and text, Inspiration 2.0 won't do a thing for you. But if you think in terms of relationships and their organization, this program must be tried to be appreciated.

I've spent the last month doing just that. In fact, I've fallen into its organiza-

tional metaphors so easily that it's sometimes hard to go back to a more traditional outliner like More or Acta. I'm now using Inspiration 2.0 in my introductory programming class (in which I teach HyperCard 2.0 and HyperTalk) to help my students invent and categorize algorithms and get them into a form that will suggest code to them.

One way Inspiration 2.0 does that is by letting them come at both ends of a programming problem at the same time. My students learn about top-down design and bottom-up testing right from the start, but one thing they often miss is the interaction between the two. Because Inspiration 2.0 can present your information either as a free-form chart or as an outline, you can use the chart to plan your global program segments (top-down design) and the outliner to punch out the code necessary for each segment (bottom-up testing).

In a flowchart created with Inspiration 2.0 (shown at left), the general algorithm for factoring a polynomial has been outlined. To code individual segments of this algorithm, you switch to Inspiration's outline view, where you can bang out and reorganize all the code statements (see listing 1). This sort of view change doesn't really scream to be noticed, but it's exactly the kind of subtle capability that makes Inspiration 2.0 so useful.

Inspiration 2.0 isn't flashy, it isn't full of multimedia bells and whistles, it doesn't fit any traditional business computing category, and it's not expensive. Yet it's exactly the kind of software that made the Mac a ground-breaking machine and keeps me hoping for better things. It's one more reason why the Mac can easily play the power, performance, and usable facility game with any desktop computer.

ITEMS DISCUSSED

Inspiration 2.0\$199
Ceres Software
2520 Southwest Hamilton St.
Portland, OR 97201
(503) 245-9011
Inquiry 1146.

Don Crabb is the director of laboratories and a senior lecturer for the computer science department at the University of Chicago. He is also a contributing editor for BYTE. He can be reached on BIX as "decrabb."

BYTE

Reprinted with permission from the December 1990 edition of BYTE magazine.
Copyright ©1990 by McGraw-Hill, Inc., New York, 10020. All rights reserved.



INSPIRATION AT THE YEAR'S END

The Mac markets changed during 1990, and new markets are developing

How time flies. The first year of the new decade whipped by as fast as a screen update on a Mac IIx. It's time to evaluate what's happened to the computing community, including the Mac's position within it. In the IBM-compatible world, the Rest of Them have been cranking out 486 systems at higher clock speeds, and Windows 3.0 is a major improvement as far as DOS-based graphical user interfaces (GUIs) go.

Apple, after some dithering about, took muster and waged the battle of the market share on two fronts. It took the high ground in March with its introduction of the Mac IIx, the first-ever Mac clocked at 40 MHz. In October, Apple began the blitzkrieg on the low end with three new low-cost Macs. Just as important, the company slashed prices on the existing models.

Apple's attack plan so far has been impressive, but it's not without flaws. The Mac IIx's much-touted SCSI DMA will not work with anything but A/UX. System 7.0 still hasn't seen the light of day. The features on the low-cost Macs are so-so, including their pricing. If Apple is indeed in the market-share battle for the long haul, I wonder whether it will have the nerve to reduce prices further if the situation warrants it. In other words, we've gone through another year of some improvements to the Mac and its software, but we're still waiting for the next Mac revolution. I'm getting pretty itchy waiting for it to come.

The one place I see a difference is not in the volleys from the big guns over-



head, but when I look in the trenches where the small Mac developers hunker down. Try as we might, we just can't kill these little guys. We entice them with promises of greater profits (Windows 3.0), hotter development systems (Next-Step), and other baubles, but they stick to the Mac like glue.

Whatever the reason, it's these folks who make the most interesting Mac software today, and who typically define new software categories. These developers think about more than just adding more features to a word processor or better kerning to a page-layout system. They think about what might be possible with a Mac.

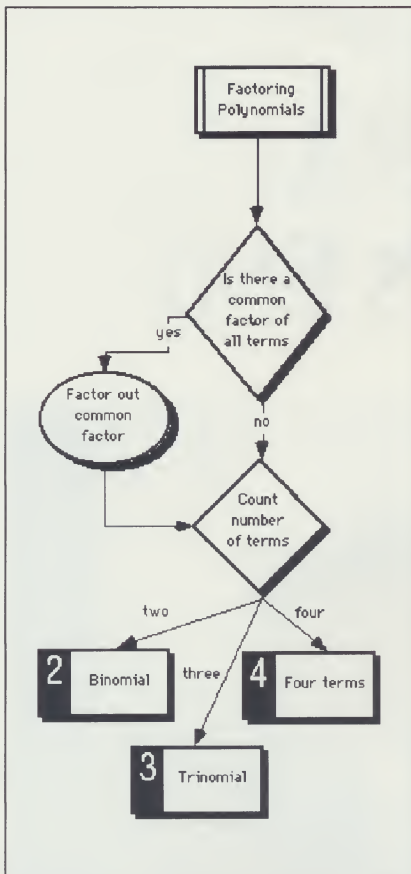
The Mac Gets Inspired

Not surprisingly, one of the newer Mac products I've found is called Inspiration 2.0. It's from a tiny family-owned company called Ceres Software in Portland, Oregon, and it's been around for a year

now, largely ignored. I spent a pleasant time conversing with its creator, Don Helfgott. He calls his program a thought processor, but that doesn't do it justice.

Inspiration 2.0 takes the Mac's ability to give you a consistent and meaningful GUI and lets you brainstorm, cranking out ideas by the bushel basket. It then organizes the ideas in a way that helps you realize what, if anything, you've really invented. Inspiration 2.0 is sort of a visual diagramming and outlining tool interwoven with a simple word processor. It's the first computer aid to creative thinking I've tried that actually works—and I've tried them all.

The theory behind Inspiration 2.0 goes something like this. Databases, word processors, and spreadsheets store and access information in ways that don't help us relate information more clearly, because they fail to distinguish between right-brain thinking (i.e., creativity, images, comprehension, imagination,



Inspiration 2.0 combines visual diagramming and outlining with a simple word processor. This Inspiration flowchart outlines the general algorithm for factoring a polynomial, as shown in listing 1.

Listing 1: This listing was derived from the Inspiration flowchart shown in the figure.

```

Factoring polynomials
  Is there a common factor of all terms?
    Count number of terms
      Binomial
        Difference of two squares
           $a^2 - b^2 = (a + b)(a - b)$ 
        Sum of two cubes
           $a^3 + b^3 = (a + b)(a^2 - ab + b^2)$ 
        Difference of two cubes
           $a^3 - b^3 = (a - b)(a^2 + ab + b^2)$ 
      Trinomial
        Trial and error or AC method
      Four terms
        Factor by grouping
      Factor out common factor
  
```

visualization, music, and daydreaming) and left-brain thinking (i.e., logic, sequence, reasoning, and analysis). Inspiration 2.0 works by filling the gap between these two kinds of thinking.

Other programs have promised to accentuate the synergy between the right and left brains. Neil Larson's Maxthink, early versions of Dave Winer's Think-Tank outliner, ODS's Consultant, and a myriad of other shareware have all skipped through this software minefield before. None of them really pulled it off, however, because they just didn't have a good model to implement.

Inspiration 2.0 has a good model, and that's why it succeeds. Rather than trying to be all things to all thinkers, the Helfgotts have pinned down their customers—writers, entrepreneurs, creative directors, analysts, planners, educators, and consultants.

They did not make Inspiration 2.0 some kind of unwieldy general management tool, or imbue it with powerful features that its target audience will not need. In short, they didn't kitchen-sink this product by trying to make it fit the needs of a general presentation tool, a business graphics package, or a free-form charting tool.

The only thing that Inspiration 2.0 assumes is that you think visually. If you never leave the mental domain of numbers and text, Inspiration 2.0 won't do a thing for you. But if you think in terms of relationships and their organization, this program must be tried to be appreciated.

I've spent the last month doing just that. In fact, I've fallen into its organiza-

tional metaphors so easily that it's sometimes hard to go back to a more traditional outliner like More or Acta. I'm now using Inspiration 2.0 in my introductory programming class (in which I teach HyperCard 2.0 and HyperTalk) to help my students invent and categorize algorithms and get them into a form that will suggest code to them.

One way Inspiration 2.0 does that is by letting them come at both ends of a programming problem at the same time. My students learn about top-down design and bottom-up testing right from the start, but one thing they often miss is the interaction between the two. Because Inspiration 2.0 can present your information either as a free-form chart or as an outline, you can use the chart to plan your global program segments (top-down design) and the outliner to punch out the code necessary for each segment (bottom-up testing).

In a flowchart created with Inspiration 2.0 (shown at left), the general algorithm for factoring a polynomial has been outlined. To code individual segments of this algorithm, you switch to Inspiration's outline view, where you can bang out and reorganize all the code statements (see listing 1). This sort of view change doesn't really scream to be noticed, but it's exactly the kind of subtle capability that makes Inspiration 2.0 so useful.

Inspiration 2.0 isn't flashy, it isn't full of multimedia bells and whistles, it doesn't fit any traditional business computing category, and it's not expensive. Yet it's exactly the kind of software that made the Mac a ground-breaking machine and keeps me hoping for better things. It's one more reason why the Mac can easily play the power, performance, and usable facility game with any desktop computer.

ITEMS DISCUSSED

Inspiration 2.0\$199
Ceres Software
2520 Southwest Hamilton St.
Portland, OR 97201
(503) 245-9011
Inquiry 1146.

Don Crabb is the director of laboratories and a senior lecturer for the computer science department at the University of Chicago. He is also a contributing editor for BYTE. He can be reached on BIX as "decrabb."